Package: integrity (via r-universe)

May 27, 2026

Type Package

Title Assessing the Integrity and Trustworthiness of Clinical Trials Data

Version 1.0.1

Date 2026-05-19

VignetteBuilder knitr

Encoding UTF-8

Depends R (>= 4.1.0)

Imports ggplot2, dplyr, janitor, gtsummary, ggpubr, lubridate, car, rlang

Suggests knitr, pkgload, readxl, rmarkdown, testthat (>= 3.0.0), yaml

Description The integrity package implements the IPD Integrity Tool, a structured and transparent framework for evaluating the integrity of individual participant data (IPD) from randomised trials (see Hunter et al. (2024) <doi:10.1002/jrsm.1738> and <doi:10.32614/RJ-2017-008>). It supports users to identify potential issues, such as unusual data patterns, implausible values, lack of expected correlations, date violations, and inconsistencies. The package provides reproducible workflows for screening, documenting and summarising integrity concerns, and may be applied by evidence synthesists, editors, and others to determine whether a randomised trial may be considered sufficiently trustworthy to contribute to the evidence base that informs policy and practice.

License GPL-3

URL `https://github.sydney.edu.au/Charles-Perkins-Centre-Data-Science-Hub/CPCDASH0010`

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

NeedsCompilation no

Author Sol Libesman [aut, cre], Kylie Hunter [aut], David Nguyen [aut], Dario Strbenac [aut], Anne Lene Seidler [aut], Jie Kang [aut]

Maintainer Sol Libesman <`sol.libesman@sydney.edu.au`>

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libxml2-dev libssl-dev libnode-dev

Repository https://sollibesman.r-universe.dev

Date/Publication 2026-05-26 10:10:02 UTC

RemoteUrl https://github.com/cran/integrity

RemoteRef HEAD

RemoteSha 887c5c9daf994cd632746142700a3f9cced5b267

Contents

`.differential_variability` *Check Variability Between Intervention and Control Groups*

Description

Internal function documentation for developers. Levene's test for differential variability.

Usage

```
.differential_variability(dataset_subset, intervention, alpha)
```

Arguments

`dataset_subset`	A `data.frame` of clinical trial data subset to only numeric columns.
`intervention`	Column name of intervention indicator.
`alpha`	p-value signficance threshold.

Value

One-row `data.frame` with a Pass or Fail indicator.

Examples

```
library(readxl)
examplePath <- system.file("extdata", "dataset.xlsx", package = "integrity")
dataset <- read_excel(examplePath)
library(yaml)
infoPath <- system.file("extdata", "variables.yaml", package = "integrity")
info <- read_yaml(infoPath)
dataset <- integrity:::.prepare_data(dataset, info)
numeric_columns <- info$baseline$numeric
dataset_subset <- dataset[, c(numeric_columns, info$intervention)]
integrity:::.differential_variability(dataset_subset, info$intervention, 0.05)
```

`.imbalance_day_intervention` *Check Day of Week of Randomisation for Non-uniformity*

Description

Internal function documentation for developers. Dates are converted into days of the week and tested for association to intervention status using `chisq.test`.

Usage

```
.imbalance_day_intervention(dataset, intervention, intervention_date, unexpected, alpha)
```

Arguments

`dataset`	A `data.frame` of clinical trial data.
`intervention`	Column name of column storing intervention status indicator.
`intervention_date`	Column name of column storing intervention date.
`unexpected`	List of elements specifying implausible values. Names of list are column names. One must be `"days"`.
`alpha`	p-value signficance threshold.

Value

A list of length two. `check_table`: One-row `data.frame` with a Pass or Fail indicator. `images`: Bar chart of days of week. Bars are coloured by intervention status.

Examples

```
library(readxl)
examplePath <- system.file("extdata", "dataset.xlsx", package = "integrity")
dataset <- read_excel(examplePath)
library(yaml)
infoPath <- system.file("extdata", "variables.yaml", package = "integrity")
info <- read_yaml(infoPath)
integrity:::.imbalance_day_intervention(dataset, info$intervention, info$enrollment$randomisation,
                                        info$unexpected, 0.05)
```

`.implausible_values` *Check Variables for Implausible Values*

Description

Internal function documentation for developers. Each column is checked for violations.

Usage

```
.implausible_values(dataset, participantID, unexpected, enrollment)
```

Arguments

`dataset`	A `data.frame` of clinical trial data.
`participantID`	Column name of column storing participant IDs.
`unexpected`	List of elements specifying implausible values. Names of list are column names
`enrollment`	Column name of column storing enrollment dates.

Value

A `data.frame` with one row for each violation or one row with Pass if no rows violated the check.

Examples

```
library(readxl)
examplePath <- system.file("extdata", "dataset.xlsx", package = "integrity")
dataset <- read_excel(examplePath)
library(yaml)
infoPath <- system.file("extdata", "variables.yaml", package = "integrity")
info <- read_yaml(infoPath)
integrity:::.implausible_values(dataset, info$participantID, info$unexpected, info$enrollment)
```

`.prepare_data` *Check clinical Data Matches its Data Specification*

Description

Internal function documentation for developers. Firstly, the function checks all expected variables are present as column names. Then, it converts any columns defined as categorical to factors. Finally, it removes any columns that have all missing values.

Usage

```
.prepare_data(dataset, info)
```

Arguments

`dataset`	A `data.frame` of clinical trial data.
`info`	A named list of column names corresponding to different aspects of the clinical trial. See the vignette for detailed requirements.

Value

If no missing colums, a `data.frame` that has been filtered for columns containing all missing values.

Examples

```
library(readxl)
examplePath <- system.file("extdata", "dataset.xlsx", package = "integrity")
dataset <- read_excel(examplePath)
library(yaml)
infoPath <- system.file("extdata", "variables.yaml", package = "integrity")
info <- read_yaml(infoPath)
integrity:::.prepare_data(dataset, info)
```

`.repeating_baseline` *Check Baseline Variables for Repetition*

Description

Internal function documentation for developers. Essentially a wrapper around `get_dupes` of janitor.

Usage

```
.repeating_baseline(dataset_subset, type = c("across", "within", "across_rare"))
```

Arguments

`dataset_subset`	A `data.frame` of clinical trial data subset to only the baseline variables.
`type`	If `"across"`, across all baseline variables. If `"within"`, within each baseline variable. If `"across_rare"`, across the baseline variables but only for participants who had a rare outcome.

Value

A `data.frame` with one row for each repetition or just one row reporting Pass status for the check.

Examples

```
library(readxl)
examplePath <- system.file("extdata", "dataset.xlsx", package = "integrity")
dataset <- read_excel(examplePath)
library(yaml)
infoPath <- system.file("extdata", "variables.yaml", package = "integrity")
info <- read_yaml(infoPath)
dataset_subset <- dataset[, unlist(info$baseline)]
integrity:::.repeating_baseline(dataset_subset)
```

`.terminal_digits` *Check Terminal Digits of Numerical Variables for Non-uniformity*

Description

Internal function documentation for developers. Creates a distribution plot of terminal digits

Usage

```
.terminal_digits(dataset_subset)
```

Arguments

`dataset_subset`	A `data.frame` of clinical trial data subset to only numeric columns.

Value

A `ggplot2` plot.

Examples

```
library(readxl)
examplePath <- system.file("extdata", "dataset.xlsx", package = "integrity")
dataset <- read_excel(examplePath)
library(yaml)
infoPath <- system.file("extdata", "variables.yaml", package = "integrity")
info <- read_yaml(infoPath)
```

```
numeric_columns <- info$baseline$numeric
dataset_subset <- dataset[, unlist(info$baseline)]
integrity:::.terminal_digits(dataset_subset)
```

`.unexpectedly_uncorrelated` *Check Pairs of Variables Expected to be Correlated*

Description

Internal function documentation for developers. Essentially, `cor.test`.

Usage

```
.unexpectedly_uncorrelated(dataset_subset, pairs, alpha)
```

Arguments

`dataset_subset`	A `data.frame` of clinical trial data subset to numeric columns.
`pairs`	List of elements, each of length two. The elements are column names.
`alpha`	p-value signficance threshold.

Value

A list of length two. `check_table`: One-row `data.frame` with a Pass or Fail indicator for each variable pair. `images`: Scatter plots.

Examples

```
library(readxl)
examplePath <- system.file("extdata", "dataset.xlsx", package = "integrity")
dataset <- read_excel(examplePath)
library(yaml)
infoPath <- system.file("extdata", "variables.yaml", package = "integrity")
info <- read_yaml(infoPath)
integrity:::.unexpectedly_uncorrelated(dataset, info$correlated, 0.05)
```

`read_metadata_excel` *Read Dataset Metadata from an Excel Template*

Description

Reads a metadata workbook template and converts it into the named list structure required by `run_checks()`.

Usage

```
read_metadata_excel(path, sheet = 1)
```

Arguments

`path`	Path to an Excel workbook containing metadata rows.
`sheet`	Sheet name or position to read. Default: 1.

Value

A named `list` suitable for the `info` argument of `run_checks()`.

Examples

```
if(interactive())
{
  example_path <- system.file("extdata", "variables_template.xlsx", package = "integrity")
   dataset_info <- read_metadata_excel(example_path)
   names(dataset_info)
}
```

`read_metadata_r` *Read Dataset Metadata from an R Script*

Description

Sources an R script template and returns the metadata list required by `run_checks()`.

Usage

```
read_metadata_r(path, object_name = "dataset_info")
```

Arguments

`path`	Path to an R script containing a metadata object.
`object_name`	Name of the object to return from the R script. Default: `"dataset_info"`.

Value

A named `list` suitable for the `info` argument of `run_checks()`.

Examples

```
if(interactive())
{
  example_path <- system.file("extdata", "variables_template.R", package = "integrity")
  dataset_info <- read_metadata_r(example_path)
  names(dataset_info)
}
```

`run_checks` *Run a Suite of Integrity Checks Based on Dataset Annotation*

Description

Depending on the characteristics of the variables, some test may be skipped if the data type required for the test is not present.

Usage

```
run_checks(dataset, info, alpha = 0.05)
```

Arguments

`dataset`	A `data.frame` of clinical trial data.
`info`	A named list of column names corresponding to different aspects of the clinical trial. See the vignette for detailed requirements.
`alpha`	Default: 0.05. For checks which use a statistical test, the p-value threshold at which to report a failure.

Value

A `list` with the element named `"check_table"` having the table of passes and fails, the element named `"detail_tables"` storing additional per-variable results for selected checks, the element named `"images"` storing ggplot2 plots and the element named `"summary_table"` having an overview table of the baseline and outcome variables split by intervention.

Examples

```
if(interactive())
{
  library(readxl)
  examplePath <- system.file("extdata", "dataset.xlsx", package = "integrity")
  dataset <- read_excel(examplePath)
  library(yaml)
  example_path <- system.file("extdata", "variables.yaml", package = "integrity")
```

```
  dataset_info <- read_yaml(example_path)
  result <- run_checks(dataset, dataset_info)
  names(result)
}
```

Index